Ensembles

BA810: Supervised Machine Learning

Nachiketa Sahoo

Recap

Steps of a Machine Learning Project

1. Load, explore data, training-test split
2. Consider cleaning and transformations
    1. Imputation, feature selection
3. Create pipelines
4. Evaluate **predictive models**
    1. Linear models, k-NN, SVM, DTree, …
5. Finetune hyperparameters of the most promising model
    1. Grid/Random/Bayesian search
6. Estimate error on unused test-data

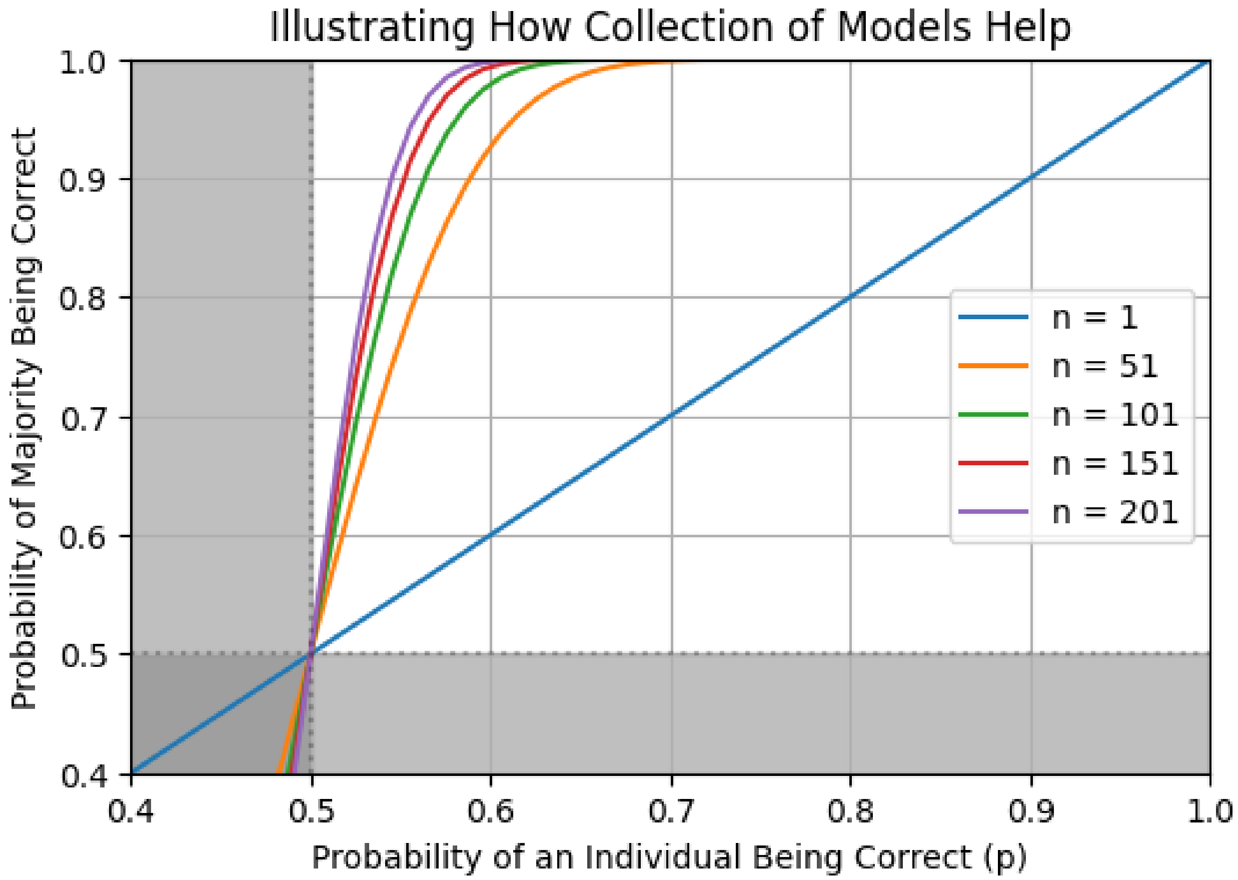
Decision Trees

- Learning: partition data into homogenous regions (leaves)
    - Prune by charging a cost per leaf/region
    - Use the cost rate with the smallest CV error
- Predict: check feature values of a test record and follow the branches of the tree until we reach a leaf node with prediction.
- Pros: Easy to explain, can handle non-linear boundaries easily, fast to learn
    - High variance (compared to other methods such as logistic/linear regression)
    - Can we make the predictions more stable?

Bagging

Bootstrap Aggregating

- Train multiple trees/models (*weak learners*) and combine their votes
  - If standard deviation of a random variable is 10, what is the standard deviation of average of 100 such variables?
  - Even if each classifier has 51% chance of being correct, majority of 1000 of them will be correct ≈73% times.
  - Only if they are independent! All predicting right/wrong together doesn't help.



Bagging
Bootstrap Aggregating

- **Bagging**
  - Draw B random samples with replacement
  - Train B decision trees (or any other predictive model)
  - Combine votes by predicting average (regression) or mode (classification)
  - Better, but still not very independent in practice (a few strong attributes for a dataset dominate)
- **Random Forest**: Bagging considering a subset of features for each *tree* node splitting
  - Number of trees hyper parameter is not critical after certain level; too many don't overfit
  - But those that control tree sizes (min leaf size, max depth, etc.) matter.
    - Cross Validation (to choose) is computationally costly we are learning potentially hundreds of trees.
- Out-of-bag error (cheaper to compute substitute for cross validation)
  - Bootstrap with replacement leaves some records out of training for each model (≈37%)
  - Predict for each record using the collection of models that didn't train on it, average the errors

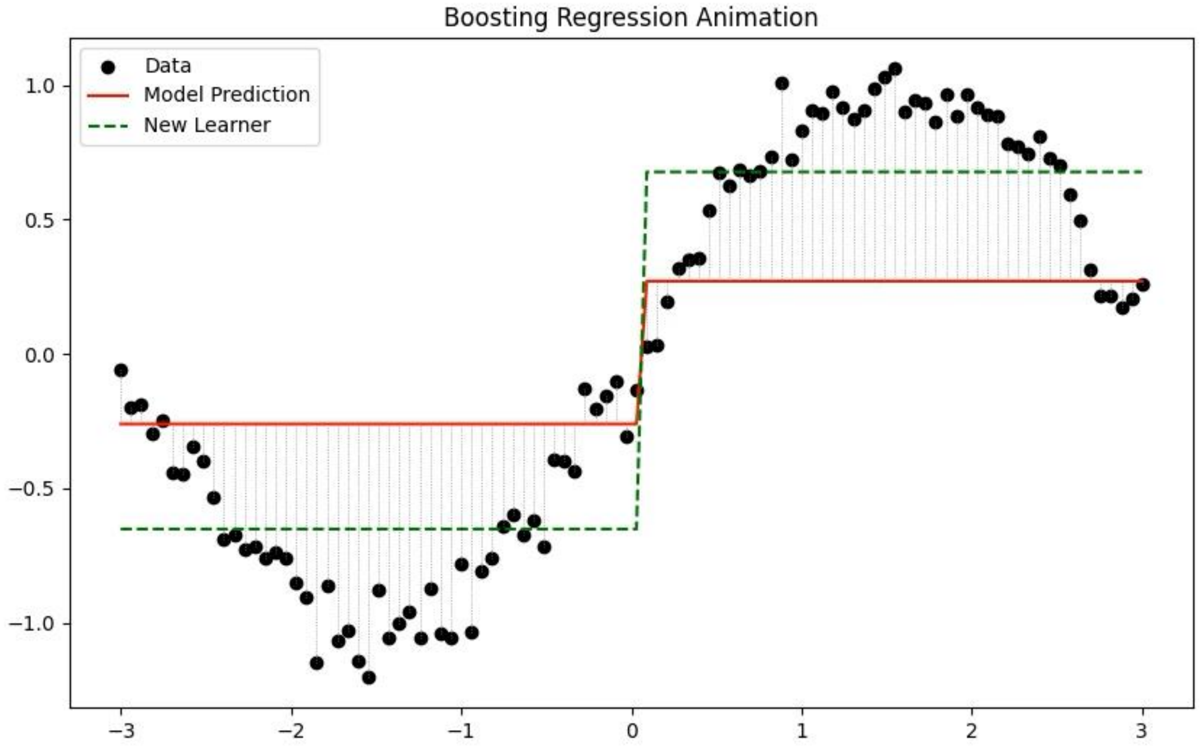
Boosting

Sequentially/smartly construct models to predict better where the errors are higher

For Regression

1. Initialize f to predict 0 for all x, thus residual is same as data
2. Repeat B times
   1. Fit a *weak learner* to the residuals: the points with largest error provide a higher influence to learning
   2. Reduce residuals by $\lambda \times$ prediction

   (i.e., add new model to the old)
3. Output $\lambda \times$ sum of B *weak learners* (variations exist)
   - λ (learning rate) is a hyperparameter



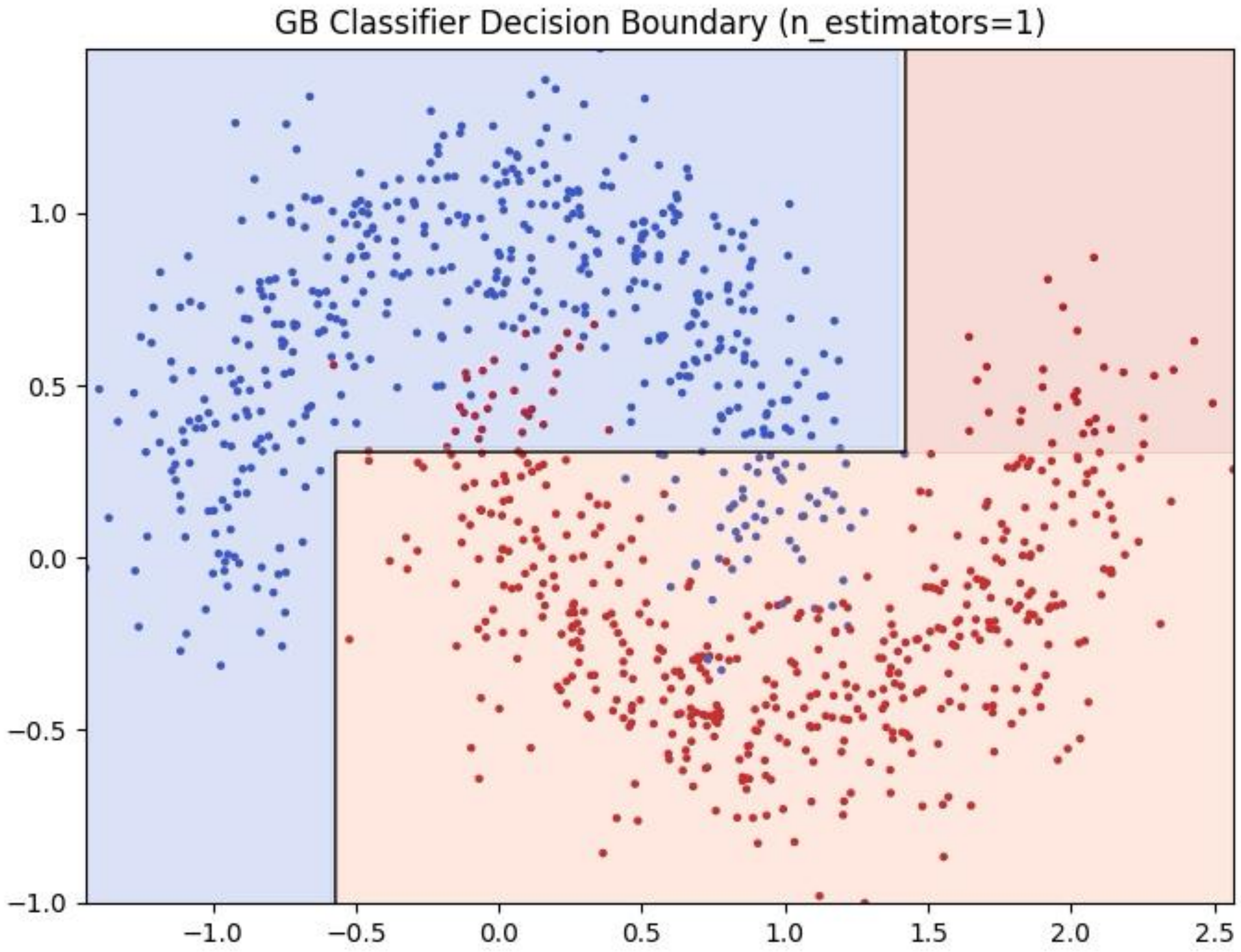
Boosting

Sequentially/smartly construct models to predict better where the errors are higher

Similar for classification—give higher weights to points that accurate models have misclassified.

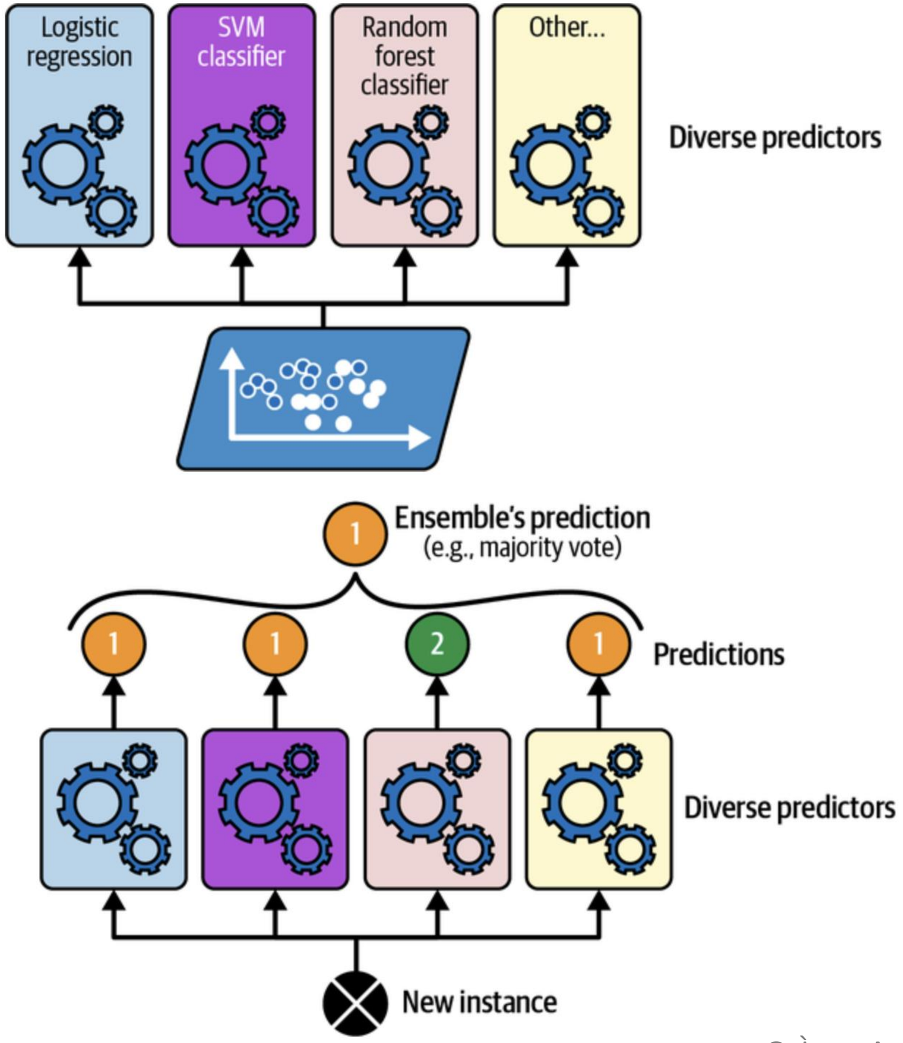
(See HOML textbook Ch 7)

- XGBoost: regularizes weak learner trees, penalizes for complexity, plus optimization to handle large data



Combine Distinct Methods by Taking a Vote

- Train multiple independent predictive models in parallel, then
  - For regression: average
  - For classification: take a vote on their prediction at each test point
    - Hard vote: count the votes for each class
    - Soft vote: weigh each method's the vote towards all classes by its probability/confidence, add across predictors, predict the class label with maximum weight
  - Voting Classifiers and Regressor
- Collaborating teams don't have to reveal their secret algorithms to each other!

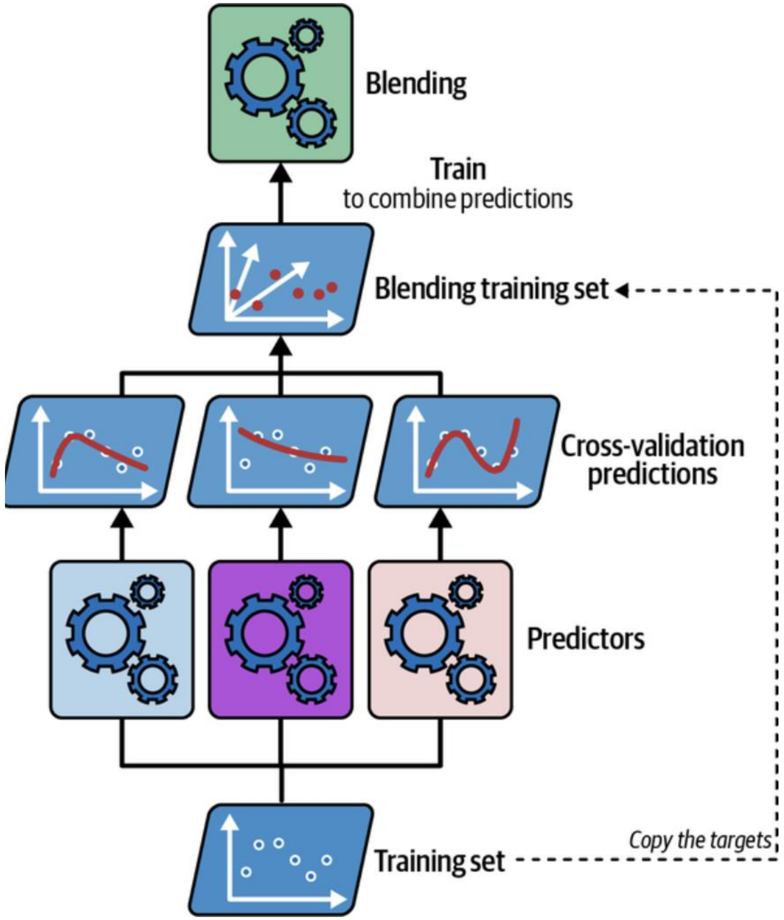


Gelron, A. (2019)

Combine Distinct Methods by Taking a Vote

- Perhaps one model is more accurate than the other; we'd like to give it more weight
- **Stacking**:
  1. Get cross validation prediction from each member model → one column per model and one row per training data
  2. Train a final/meta model with CV predictions as input and actual as output, on training data
  3. Train and keep new copies of member models on entire training data

  Predict using metal model from step 2 using member model predictions from step 3 as input.
    - Could be better than using only the strongest model because other models can perform better in some areas
    - Netflix prize teams (who tied with the winner) used this.
    - More details in HOML Ch 7.



Geìron, A. (2019)

Ensemble Summary

- Combine multiple estimated models to predict better
  - Any one prediction can be too sensitive to noise, but their average or mode is likely to be more stable
- Reducing correlation in prediction of different models motivates many methods
  - Bagging: bootstrap to create different training data to create different models
  - Random Forest: Bagging with trees, but at split points consider random subset of features
  - Boosting: Instead of learning independent models, learn sequentially to fit new models to predict better in areas with high errors
- Voting
  - Can be applied to any group of arbitrary models (don’t have to be of same type)
    - Learn many, use mean/mode of the prediction
  - Teams can keep their secrets!
  - Stacking: gives more weight to the model that predicts better
    - Learn the weights as a supervised learning task